Robotics and Autonomous Systems 87 (2017) 177–187

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A comparison of continuous and discrete tracking-error model-based predictive control for mobile robots

Igor Škrjanc, Gregor Klančar *

Laboratory of Modelling, Simulation and Control, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia

HIGHLIGHTS

- We present a new continuous tracking-error model-based predictive control algorithm for mobile robots.
- Comparisons are made to our previous work with discrete design.
- Better performance, the design parameters are insensitive to the sampling time.
- Approach enables a non-uniform sampling which is natural in many applications.

ARTICLE INFO

Article history:
Received 24 December 2014
Received in revised form 2 September 2016
Accepted 15 September 2016
Available online 12 October 2016

Keywords:
Continuous model-based predictive control
Trajectory tracking
Mobile robot

ABSTRACT

Model-based predictive control approaches can be successfully applied to the trajectory tracking of wheeled mobile-robot applications if the process nonlinearity is considered, if real-time performance is achieved and if assumptions made in the control-law design are met when applied to a particular process. In this paper, continuous tracking-error model-based predictive control is presented. The controller's optimal actions are obtained from an explicit solution of the optimization criteria, which enables fast real-time applications. Due to its design in continuous time, its usage is not limited to the uniform sampling restrictions of a host computer, as is usually the case in discrete time design. Therefore, better performance is obtained in applications with non-uniform sampling, which is natural in many situations due to imperfect sensors, mismatched clocks, nondeterministic control delays or because of the unknown time of the pre-processing. The controller-design parameters are insensitive to the sampling time period, which contributes to simpler applications and greater robustness of the controller.

1. Introduction

Wheeled-robot motion control is important in practical applications as well as being an important research problem. Different control laws were proposed for driving mobile robots with differential kinematics [1]. The motion control of such robots can be carried out as a point stabilization [2,3] or as trajectory tracking [4–7]. Trajectory tracking appears to be more natural for mobile-robot drives with nonholonomic constraints.

A very common and frequently used nonlinear controller design is that which first appears in [7–9]. It is designed in a Lyapunov frame and guarantee asymptotic stability. This controller structure has motivated many researchers to include their modifications, such as an adaptive upgrade in [10], a fuzzy extension in [11], an input–output linearization in [12], a saturation-constraint feedback in [13], a combined control and observer design in [14], and many others. In [7] a dynamic feedback linearization for a flat system output is described, which results in a more robust design and does not require any orientation measurements. In [15] a Lyapunov analysis is used to design a nonlinear control law that is asymptotically stable and overcomes the common discontinuity problem in the orientation error. Control of many commercial robots can be done considering kinematic model only because they already have internal control handling robot dynamics. If this is not the case a dynamic compensator [5] should be implemented before applying kinematic control.

Approaches of nonlinear MPC (Model Predictive Control) for tracking in mobile robots are rare [16], with the earliest papers being [17,18] and [19]. In these applications the computational burden was prohibitive in fast, real-time applications. Later, several real-time implementations followed in [16,20] and [21] where optimized numeric search approaches are applied to solve the MPC optimization problem. An analytical solution of the MPC problem for mobile robots is proposed in [22], which enables fast and simple real-time implementations. Several model predictive approaches

* Corresponding author. Fax: +386 1 4264631
E-mail address: gregor.klancar@fe.uni-lj.si (G. Klančar).

http://dx.doi.org/10.1016/j.robot.2016.09.016
0921-8890/

apply linearization to obtain computationally more efficient solutions that are valid near the operating point. If environment disturbances are high a nonlinear predictive approaches [19] or robust solutions should be used. Robust predictive approach which can handle several disturbances that usually can appear in outdoor applications is suggested by [23]. In these model predictive approaches a discretization of the nonlinear or linearized system model is required, resulting in a discrete control law. However, the discretization to the required periodic sampling is only approximate, especially if the period of the sampling is not exact. Therefore, it introduces some systematic error in the control-law calculation. Discrete control approaches used on continuous-time plant lose the information of intersample operation of continuous process [24]. A study how variable sampling period due to random delays influences stability of mobile robot control is performed in [25]. Control of nonlinear system under variable sampling have been investigated in [26] applying a fuzzy control approach.

In this work a continuous model-based predictive control is proposed. The control law has a similar structure to the discrete MPC in [22] and differs mainly in terms of a design that is made in continuous space. The main novelties of the proposed approach, with respect to our previously published approach [22], are as follows. The predictive control law is designed in a continuous space, which means that discretization of the tracking-error dynamics is not needed. Better, or at least equal, trajectory tracking results are obtained because the error due to the discretization is not present in the control law. A greater robustness of the control-law design parameters, such as the time horizon and the desired control law dynamics, to sampling-time variations is obtained. The design parameters are insensitive to the used sample time, which is not the case with the discrete design. This means that continuous model predictive control can be realized in non-equidistant sampling cases where better trajectory-tracking results are obtained compared to the results of discrete model predictive controls.

The rest of the paper is organized as follows. In Section 2 the continuous model predictive control law is derived for a mobile robot with differential kinematics. Comparisons of the simulation results between the proposed continuous model predictive control and the discrete model predictive control are given in Section 3. The experimental results and comparisons are presented in Section 4, and the conclusions are drawn at the end.

2. Trajectory-tracking problem

In this section a continuous tracking-error model-based control algorithm is explained, which is applied to the mobile robot with differential drive kinematics as follows

$$\dot{q}(t) = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \tag{1}$$

where $v(t)$ and $\omega(t)$ are the tangential and angular velocities of the mobile robot, q is $[x, y, \theta]$ and $\dot{q}$ is their derivative. Obtained control results in the following can be extrapolated to other mobile platforms such as very often used Ackermann type. Model (1) can be applied to Ackermann using transformations $v(t) = v_s(t)\cos\alpha(t)$ and $\omega(t) = \frac{v_s(t)}{d}\sin\alpha(t)$ where α is the steering angle, v_s velocity of the steering wheel and d the distance among the steering and the rear wheels [27].

In the trajectory-tracking problem the control task is to follow the given reference trajectory. This can be solved by a nonlinear feedback or a smooth linear feedback designed for a linearized system around the trajectory ([2,28–30] and [9]). To achieve asymptotic stability of the nonholonomic system (1) a time-varying feedback is needed [31].

The reference trajectory $x_r(t)$, $y_r(t)$ is achievable for a differential drive if it is twice differentiable and does not come to a stop ($\dot{x}_r^2(t) + \dot{y}_r^2(t) \neq 0$). If the latter is true, the feedforward controls can be calculated from the reference trajectory. The tangential feedforward velocity $v_r(t)$ is obtained by

$$v_r(t) = \left(\dot{x}_r^2(t) + \dot{y}_r^2(t)\right)^{\frac{1}{2}} \tag{2}$$

and the angular feedforward velocity $\omega_r(t)$ is obtained from time derivative of the tangent orientation of the reference trajectory $\theta_r(t) = \arctan\frac{\dot{y}_r(t)}{\dot{x}_r(t)}$ as follows

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)}. \tag{3}$$

The feedforward control action is only applicable if the robot is perfectly described by the kinematic model and if no disturbances and initial posture errors are present. In practice, the feedforward control action is supplemented by a suitable feedback control law.

2.1. State tracking-error kinematics

The state trajectory tracking error $e(t)$ defined in the robot coordinate frame is obtained using

$$e(t) = \begin{bmatrix} e_x(t) \\ e_y(t) \\ e_\theta(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & \sin\theta(t) & 0 \\ -\sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} (q_r(t) - q(t)). \tag{4}$$

From the kinematics (1) state tracking error (4) and supposing that the imaginary reference robot has the same kinematics (1), the following model results

$$\dot{e}(t) = \begin{bmatrix} \cos e_\theta(t) & 0 \\ \sin e_\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r(t) \\ \omega_r(t) \end{bmatrix} + \begin{bmatrix} -1 & e_y(t) \\ 0 & -e_x(t) \\ 0 & -1 \end{bmatrix} u(t) \tag{5}$$

where $u(t) = [v(t) \quad \omega(t)]^T$ stands for the control vector. Robot control is obtained by combining the feedforward and feedback control actions

$$u(t) = u_f(t) + u_b(t) \tag{6}$$

where $u_f(t) = [v_r(t)\cos e_\theta(t) \quad \omega_r(t)]^T$ is the feedforward part and $u_b(t) = [v_b(t) \quad \omega_b(t)]^T$ is the feedback part.

By inserting relation (6) into Eq. (5), the nonlinear state tracking-error kinematics is obtained as follows

$$\dot{e}(t) = \begin{bmatrix} 0 & \omega_r(t) & 0 \\ -\omega_r(t) & 0 & v_r(t)\dfrac{\sin e_\theta(t)}{e_\theta(t)} \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} -1 & e_y(t) \\ 0 & -e_x(t) \\ 0 & -1 \end{bmatrix} u_b(t). \tag{7}$$

For the purposes of continuous model-predictive control a linearization of (7) around the reference trajectory (desired operating point: $e_x(t) = e_y(t) = e_\theta(t) = 0$, $v_b(t) = \omega_b(t) = 0$) is performed to obtain a linear continuous model

$$\dot{e}(t) = \begin{bmatrix} 0 & \omega_r(t) & 0 \\ -\omega_r(t) & 0 & v_r(t) \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_b(t) \tag{8}$$

whose compact form is defined as follows $\dot{e}(t) = A(t)e(t) + Bu_b(t)$. This compact linear form will be used in the subsequent text to derive explicit control law. Note, however that linear model is only valid in vicinity of the operating point (zero error in (8)) and the control performance using linear model may not be as expected in case of large control errors. Here controller that forces error towards zero is designed therefore the linear model is acceptable choice.

2.1.1. Prediction of errors

The prediction of a certain error component is obtained with a *Taylor series* expansion as follows

$$e_i(t+\tau) = e_i(t) + \sum_{k=1}^{n_e} e_i^{(k)}(t)\frac{\tau^k}{k!}, \quad i = 1, \ldots, n, \tag{9}$$

where $e_i^{(k)}(t)$ defines the kth time derivative of the variable $e_i(t)$ as follows

$$e_i^{(k)}(t) = \frac{d^k e_i(t)}{dt^k} \tag{10}$$

and n_e defines the order of the prediction, i.e., the order of the derivatives in the series expansion and τ defines the time of the prediction. The error of this approximation depends on the order n_e and the time of the prediction τ.

Eq. (9) is then rewritten in the form as follows

$$e_i(t+\tau) = e_i(t) + \left[\tau \;\; \frac{\tau^2}{2!} \;\; \cdots \;\; \frac{\tau^{n_e}}{n_e!}\right]\left[e_i^{(1)}(t)\; e_i^{(2)}(t) \;\; \cdots e_i^{(n_e)}(t)\right]^T. \tag{11}$$

The kth derivative of the error vector is then similarly, taking into account Eq. (8), written in the following form

$$e^{(k)}(t) = A^{n_e}(t)e(t) + \left[A^{n_e-1}(t)B \;\; A^{n_e-2}(t)B \;\; \cdots \;\; B\right]u_b^*(t) \tag{12}$$

and $u_b^*(t)$ stands for $u_b^*(t) = \left[u_b(t)^T \;\; u_b^{(1)}(t)^T \;\; \ldots \;\; u_b^{(n_e-1)}(t)^T\right]^T$ and has the dimension of $m \cdot n_e \times 1$ and $e(t)$ has the dimension $n \times 1$, where m stands for the input-vector dimension.

By taking into account Eqs. (9) and (12), the prediction of all the error variables will be given as follows

$$e(t+\tau) = e(t) + T_e\begin{bmatrix} e^{(1)} \\ e^{(2)} \\ \vdots \\ e^{(n_e)} \end{bmatrix}, \tag{13}$$

where T_e stands for the following $n \times n \cdot n_e$ matrix

$$T_e = \left[\tau I_n \;\; \frac{\tau^2}{2!}I_n \;\; \cdots \;\; \frac{\tau^{n_e}}{n_e!}I_n\right] \tag{14}$$

and I_n stands for the $n \times n$ identity matrix.

Eq. (13) can be further, by taking into account Eq. (12), developed as follows

$$e(t+\tau) = e(t) + T_e F(t)e(t) + T_e H(t)u_b^*(t), \tag{15}$$

where $F(t)$ stands for the matrix of dimension $n \cdot n_e \times n$, defined as

$$F(t) = \left[A(t) \;\; A^2(t) \;\; \cdots \;\; A^{n_e}(t)\right]^T \tag{16}$$

and $H(t)$ stands for the matrix of dimension $n \cdot n_e \times m \cdot n_e$, defined as

$$H(t) = \begin{bmatrix} B & 0 & 0 & 0 \\ A(t)B & B & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A^{n_e-1}(t)B & A^{n_e-2}(t)B & \cdots & B \end{bmatrix}. \tag{17}$$

2.2. The reference-error model

The dynamics of the reference trajectory tracking is involved with the reference-error model, which is defined as follows

$$\dot{e}_r(t) = A_r e(t) \tag{18}$$

where A_r stands for the reference-error transition matrix defined by $A_r = a_r \cdot I_n$, where $a_r < 0$. This means that the nature of the trajectory tracking is defined by matrix A_r with the dimension $n \times n$ and diagonal elements, which defines the dynamics of the reference trajectory. At the current time instant $e_r(t) = e(t)$, while the future reference-error prediction must exponentially decrease to zero. This leads to a prediction of the reference error for the time τ ahead (similarly as in (15)), which is defined as

$$e_r(t+\tau) = e(t) + T_e F_r e(t), \tag{19}$$

where F_r stands for

$$F_r = \left[A_r \;\; A_r^2 \;\; \cdots \;\; A_r^{n_e}\right]^T. \tag{20}$$

2.3. The control law

The criterion function that is optimized to obtain the control law is now written in the matrix form as follows

$$\mathcal{J} = \int_0^{T_h} \left[\varepsilon^T Q \varepsilon + \Delta u_b^T(\tau) R \Delta u_b(\tau)\right] d\tau \tag{21}$$

where $\Delta u_b(\tau)$ stands for $\Delta u_b(\tau) = u_b(t+\tau) - u_b(t)$ and ε stands for $\varepsilon = e_r(t+\tau) - e(t+\tau)$. The prediction of the control variable $u_b(t+\tau)$ is defined as follows

$$u_b(t+\tau) = u_b(t) + T_u u_b^* \tag{22}$$

where T_u is defined as

$$T_u = \left[\tau I_m \;\; \frac{\tau^2}{2!}I_m \;\; \cdots \;\; \frac{\tau^{n_e}}{n_e!}I_m\right] \tag{23}$$

and I_m stands for the $m \times m$ identity matrix. The matrix Q defines the matrix of dimension $n \times n$ for the weighting of the tracking errors, R stands for the $m \times m$ matrix to weight the change of the input variable $u(t)$, and T_h is prediction horizon time where the criterion is calculated.

Taking into account Eqs. (15) and (19), integrating the criterion function and then calculating the derivative according to u_b^* the following is obtained

$$\begin{aligned}\frac{d\mathcal{J}}{du_b^*} =& -H^T T_Q F_r e + H^T T_Q F e + H^T T_Q^T F e \\ & - H^T T_Q F_r e + 2H^T T_Q H u_b^* + 2T_R u_b^*\end{aligned} \tag{24}$$

where the shorter notation $H = H(t), F = F(t)$ is used and where T_Q and T_R are the following constant positive definite matrices of dimensions $n \cdot n_e \times n \cdot n_e$ and $m \cdot n_e \times m \cdot n_e$

$$T_Q = \int_0^{T_h} T_e{}^T Q T_e d\tau \tag{25}$$

$$T_R = \int_0^{T_h} T_u{}^T R T_u d\tau \tag{26}$$

where T_e is defined as given in Eq. (14) and T_u is given in Eq. (23).

The matrices T_Q and T_R are symmetric and constant matrices that are independent of time.

A necessary condition the optimality is given by third *Euler–Lagrangove equation* as follows

$$\frac{\partial \mathcal{J}}{\partial u_b^*} = 0, \tag{27}$$

and the sufficient condition for the optimal solution is given by the *Legendre–Clebsch equation* as follows

$$\frac{\partial^2 \mathcal{J}}{\partial u_b^{*2}} = 2T_R \geq 0, \tag{28}$$

which follows from the fact that the matrix T_R is positive definite.

The optimal control law is then given as follows

$$u_b^*(t) = \left(H(t)^T T_Q H(t) + T_R\right)^{-1} H(t)^T T_Q \left(F_r - F(t)\right) e(t). \tag{29}$$

The optimal control variable $u_b(t)$ is given by the first m rows of the vector $u_b^*(t)$.

2.4. The order of the control variable

The control law is given by the control variable $u_b^*(t)$, which has the dimension $m \cdot n_e \times 1$. This means that the expansion of the control variable $u(t+\tau)$ is of the same order n_e as the expansion of the errors $e(t+\tau)$. Or, the series expansion of the control variable is given by $n_e - 1$ derivatives. This could lead to the singularity problem in the case of Eq. (29), where the inverse of the expression $H(t)^T T_Q H(t) + T_R$ is calculated. This problem is solved by introducing the control-variable order n_u. This means that the Taylor series expansion of the control variable is limited to n_u derivatives as follows: $u_b^*(t) = \left[u_b(t)^T \;\; u_b^{(1)}(t)^T \;\; \dots \;\; u_b^{(n_u)}(t)^T\right]^T$ and has the dimension of $m \cdot (n_u + 1) \times 1$. This also leads to the change of the matrix $H(t)$, which is now of dimension $n \cdot n_e \times m \cdot (n_u + 1)$, defined as

$$H(t) = \begin{bmatrix} B & 0 & 0 & 0 \\ AB & B & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A^{n_e}(t)B & A^{n_e-1}(t)B & \cdots & A^{n_e-n_u}(t)B \end{bmatrix} \tag{30}$$

and the change of T_u, which is now defined as follows

$$T_u = \left[\tau I_m \;\; \frac{\tau^2}{2!} I_m \;\; \cdots \;\; \frac{\tau^{n_u}}{n_u!} I_m\right] \tag{31}$$

and I_m stands for the $m \times m$ identity matrix. This also means that the matrix T_R now has dimension $m \cdot (n_u + 1) \times m \cdot (n_u + 1)$. The control variable order n_u should be smaller than the error variable order n_e.

3. Validation of the continuous MPC performance

In the following the performance and robustness of the proposed control law is validated by various simulations, considering the periodic and aperiodic sampling intervals, the noise in the sampling-period duration and the nondeterministic control delay.

The obtained results are compared to the discrete MPC (DMPC) realization [22], with the main purpose being to illustrate in which situations the use of the proposed continuous MPC (CMPC) is beneficial over the discrete design and also when it is not.

The reference trajectory for all the experiments is defined by

$$x_r(t) = 1.1 + 0.7 \sin\left(\frac{2\pi t}{30}\right), \qquad y_r(t) = 0.9 + 0.7 \sin\left(\frac{4\pi t}{30}\right)$$

where $t \in [0,\ 30]$ s. The robot starts with an initial state error according to the reference trajectory, its starting pose is $q = [1.1 \;\; 0.8 \;\; 0]^T$. The robot velocities and wheel accelerations are limited as follows: $v_{MAX} = 1$ m/s, $\omega_{MAX} = 15$ and $a_{MAX} = 3$ m/s^2.

The design parameters for the continuous MPC are as follows:

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}, \qquad R = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix},$$

$$A_r = \begin{bmatrix} -13 & 0 & 0 \\ 0 & -13 & 0 \\ 0 & 0 & -13 \end{bmatrix}.$$

The order of the prediction is $n_e = 3$, the order of the control variable is $n_u = 2$, the prediction horizon time $T_h = 4T_s$ and the diagonal element in the reference trajectory matrix is $a_r = -13$.

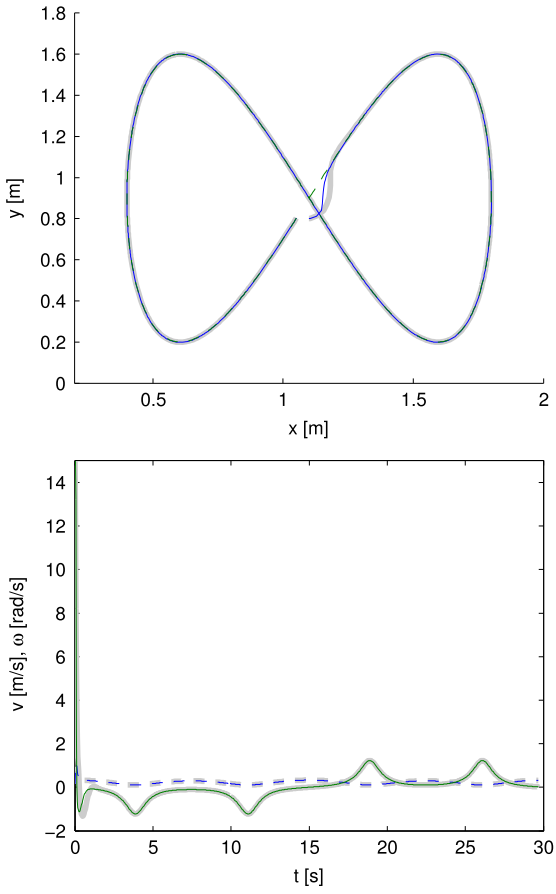

Fig. 1. Trajectory tracking of continuous (thin) and discrete (thick) model-predictive controller under ideal sampling. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

Additional parameters needed for the discrete MPC algorithm's realization that give a comparable performance to the continuous realization are as follows

$$A_{dr} = e^{a_r T_s} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.65 & 0 & 0 \\ 0 & 0.65 & 0 \\ 0 & 0 & 0.65 \end{bmatrix}$$

where the control horizon $h = 4$ and the sampling period of the control loop $T_s = 0.033$ s.

3.1. Performance simulation under ideal conditions

In this simulation we suppose that the process inputs are changed at regular sampling intervals *Ts* and no noise is present in the controlled system.

The obtained results (trajectory tracking and velocity inputs) of the continuous MPC and the discrete MPC realization are shown in Fig. 1. Both have very similar performance, because of the periodic sampling and the appropriate sampling period selection.

3.2. Robustness of the design parameters to the sampling period

An important advantage of the continuous control approach is the insensitivity of the design parameters to the sampling time. In the discrete case the design parameters of the control law depend on the sampling time. This, however, is not the case in the continuous approach. To clarify this claim the simulated sampling period is now changed to $T_s = 0.066$ s, which is two times longer than in the previous example (Fig. 1). The process inputs are changed at

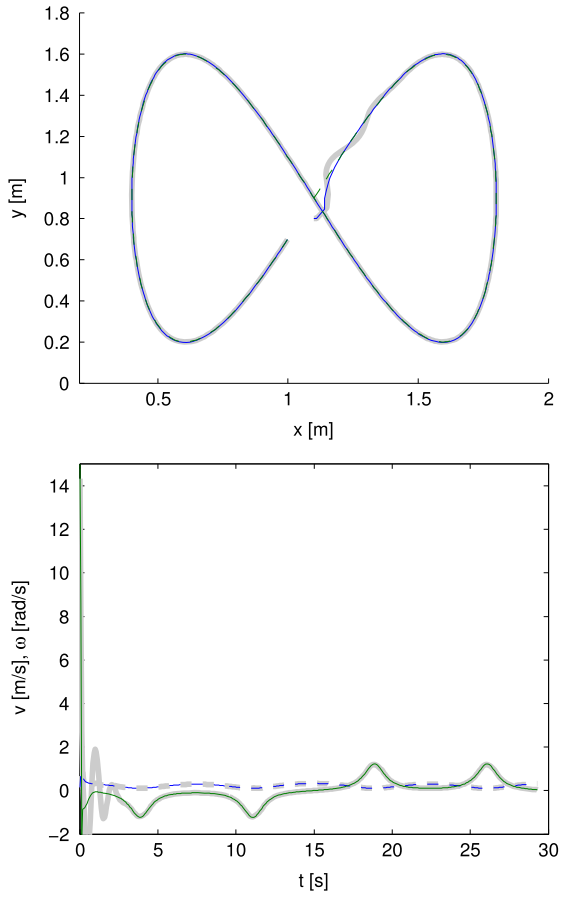

Fig. 2. Trajectory tracking of continuous (thin) and discrete (thick) model-predictive controller at double sampling time ($T_s = 0.066$). First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

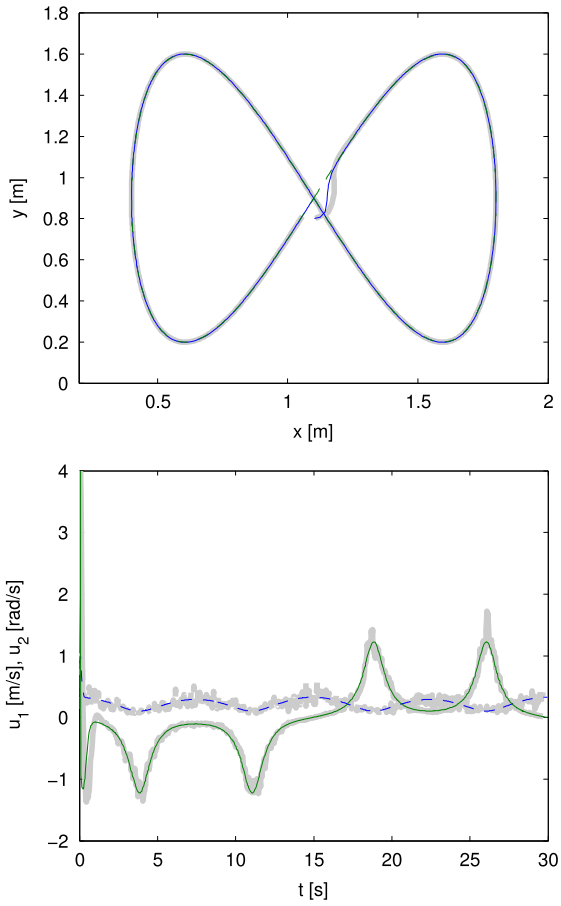

Fig. 3. Trajectory tracking of continuous (thin) and discrete (thick) model-predictive controller at variable sampling. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

regular sampling intervals $T_s = 0.066$ s, but the design parameters (A_r, Q, R, T_h) for the control law remain the same as in the previous example, so they are valid for a 0.033 s sampling period.

The results of the continuous and discrete MPC are shown in Fig. 2. The results of continuous realization are very much the same as in Section 3.1, while the performance of the discrete case is worse. The CMPC is derived in continuous space and therefore the discretization of the continuous system model is not needed, as it is in the discrete-case design. Consequentially, the design parameters of the CMPC are also not dependent on the sampling period *Ts*. Therefore, the CMPC is more robust to the sampling-period deviations (aperiodic sampling). And also in the case of the periodic sampling, the tuning of the controller parameters is not required if sampling period is changed. While DMPC controller parameters need to be tuned again when sampling time is changed.

3.3. Performance under variable sampling

Usually, the elements of the control loop (sensors, actuators, controller) are event-driven and ideal periodic sampling is rarely available [32]. However, the statistically expected value of the sampling period T_s must fulfil the criterion $0.2 \leq \omega T_s \leq 0.6$, as stated in [33], where ω is the closed-loop natural frequency.

The true sampling time T_{sTrue} is therefore nondeterministic, which in this simulation is modelled by the normal probability density function

$$p\left(T_{sTrue}\right) = \frac{1}{\sqrt{2\pi\sigma_s^2}} e^{-\frac{1}{2}\left(\frac{(T_{sTrue}-Ts)}{\sigma_s}\right)^2}$$

where $T_s = 0.033$ s is the mean value and $\sigma_s = 0.01$ s is the standard deviation. So, the process inputs are changed at time intervals T_{sTrue}, which is nondeterministic.

The results of the continuous MPC and discrete MPC are shown in Fig. 3. Due to the sample-time variation during the simulation (see Fig. 4), the noise in the control signals from the discrete MPC appears. This happens because the discrete control is not performed for regular periodic time samples and the error due to the discretization is then propagated over the whole control-law algorithm.

3.4. Performance under variable sampling and control delay

In practice, a control delay is present, which can again be modelled as a nondeterministic process. The sensor processing, controller and actuators are usually event-driven parts of the closed-loop system. So, the sensor information for the control law starts processing as soon as the raw data from the physical sensor is available, which can be assumed to be at regular sampling intervals T_s or at nondeterministic intervals T_{sTrue} (usually with quite a low uncertainty).

However, the raw sensor data need to be processed (e.g., SLAM in mobile robotics) to produce the required sensor information for the control part. This sensor processing time is time-varying and

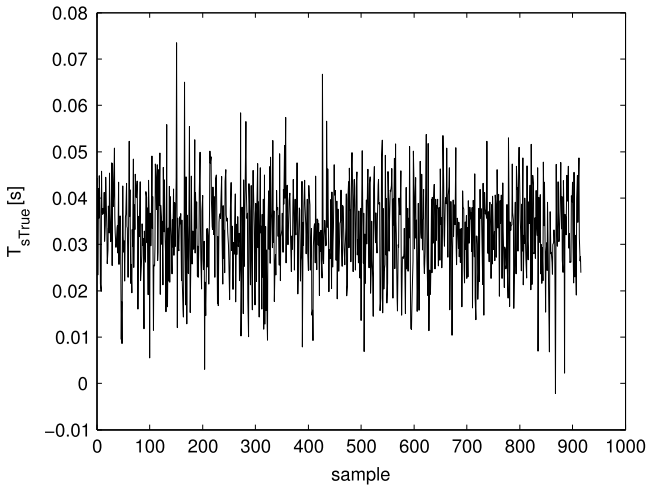

Fig. 4. Sample-time variation during the simulation.

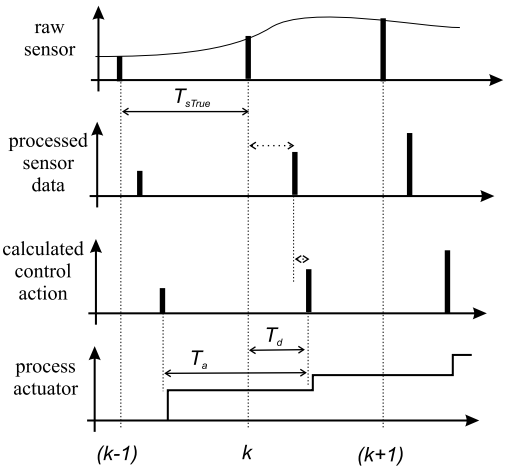

Fig. 5. Variable sampling demonstration. Raw sensor data are obtained at sample intervals T_{sTrue}, which triggers the processing algorithms to extract the required control information from the raw sensor data. The latter triggers the control algorithms and finally the control variables are communicated to the process actuators. All the mentioned algorithms and the communication cause a control delay T_d, which is usually nondeterministic as well as instants of the control inputs (actuation period T_a).

contributes to the overall time delay. The control algorithm starts when the processed sensor information is available and calculates the process input after some time delay, which is again time-varying. In mobile robotics the controller can typically have many tasks with different complexities such as path planning, obstacle avoidance, reference tracking and not all of them are always active. Finally, after the process input is calculated, the actuator produces the required input to the process, where some communication delay may be present. The overall control delay T_d is therefore nondeterministic as illustrated in Fig. 5.

The results of the continuous and discrete MPC realization are shown in Fig. 6. From the obtained results in Fig. 6 better performance is observed for the continuous MPC, where less noise appears in the control signal. The control delay disturbs both control designs, which is seen from the noise in the control signals. However, due to the variable actuation period T_a the performance of the continuous controller realization is better as it has a faster response and lower noise at the control inputs. The variable period T_a is shown in Fig. 7 and results from a nondeterministic sensor sampling and control delay.

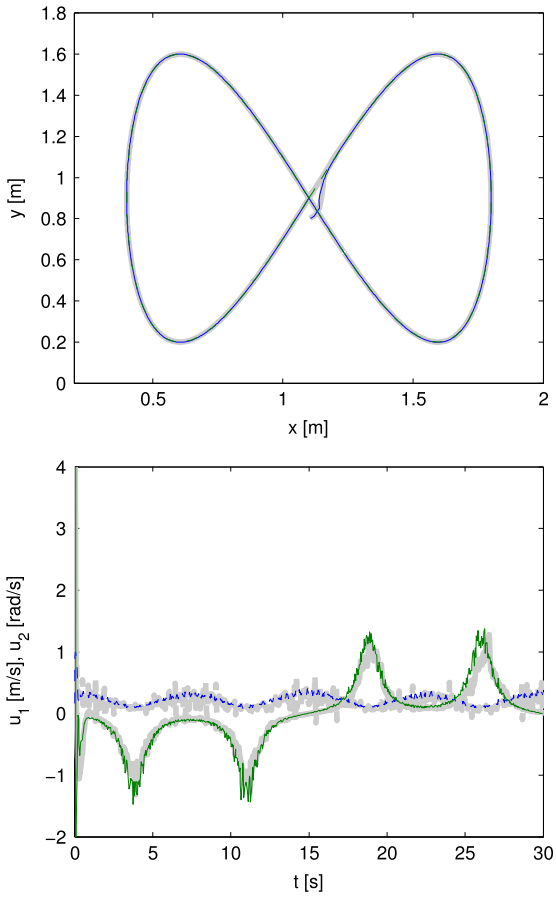

Fig. 6. Trajectory tracking of continuous (thin) and discrete (thick) model-predictive controller at variable sampling and control delay. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

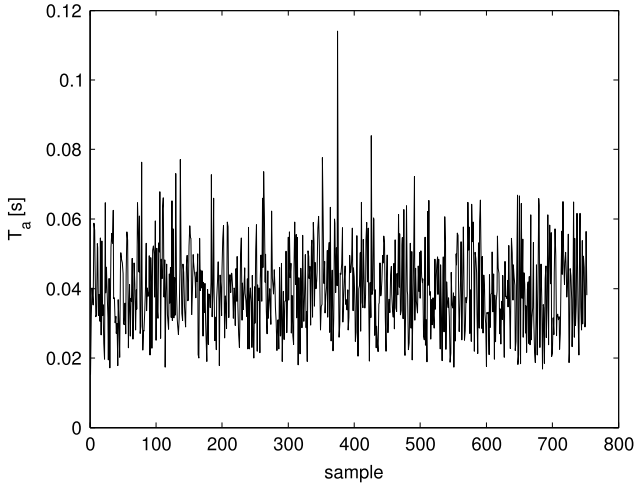

Fig. 7. Sample time of actuation T_a varies due to the nondeterministic sensor sampling and the nondeterministic process delay.

In general, the noise at the control inputs is propagated from the process output noise and also from the noise in the sampling period and control delay. The faster the controller dynamics, the larger the control noise is. However, from Fig. 6 a faster response of continuous MPC is observed at a lower control noise than in discrete MPC.

Similar conclusions can also be made for the different reference trajectories. Example of a discontinues reference trajectory, which

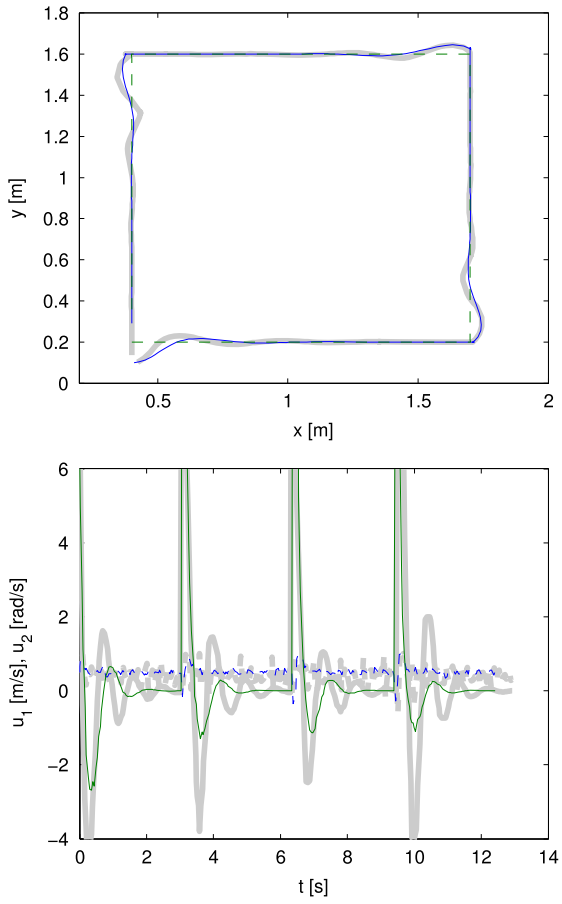

Fig. 8. Trajectory tracking with the continuous model-predictive controller at variable sampling and control delay. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

are usually the output of path planing approaches [34], is given in Fig. 8.

3.5. Quality index comparison

A comparison of the performance for the simulated scenarios from Sections 3.1–3.4 is given in Table 1. The performance is evaluated by the root-sum-square of the position error ($RSS_x = \sqrt{\sum e_x^2}, RSS_y = \sqrt{\sum e_y^2}$) and the orientation error ($RSS_\theta = \sqrt{\sum e_\theta^2}$), by the norm of the root-sum-square of the position errors ($NSS = \sqrt{(RSS_x^2 + RSS_y^2)}$) from the reference trajectory and by the standard deviations of the control inputs (σ_v, σ_ω).

In first line of Table 1 an ideal situation is compared (Section 3.1) from which it is clear that there is no noticeable difference in performance between the continuous (CMPC) and the discrete (DMPC) realizations.

In the second line of Table 1 the robustness of the control design parameters to the sampling period is tested (Section 3.2). The parameters optimized for the sampling period $T_s = 0.033$ s are used on the simulation with the sampling period $T_s = 0.066$ s. It is clear that the performance (pose tracking and control signals) of the discrete realization performs much worse than the continuous one. It has to be noted that the *RSS* and *NSS* values of the second line and the first line could not be compared due to the different number of sampling instants for the same duration of the simulation.

In the third line of Table 1 a variable sampling is simulated (Section 3.3). The main difference can be observed in the larger control-inputs noise standard deviation of the DMPC, while the CMPC performs similarly to the in ideal case. In the DMPC the noise in the control signals is caused by the noise in the sampling time, where the sampling instants are not periodic. Due to the closed-loop operation the noise in the sampling time mostly affects the control inputs, while the tracking errors are similar for the CMPC and DMPC.

In the fourth line of Table 1 a variable sampling and variable control delay is simulated. The delay affects the performance of both control designs. The variable delay also contributes to a larger variance of the time between the successive instants of the control-inputs update and, therefore increases the control-inputs noise in the DMPC.

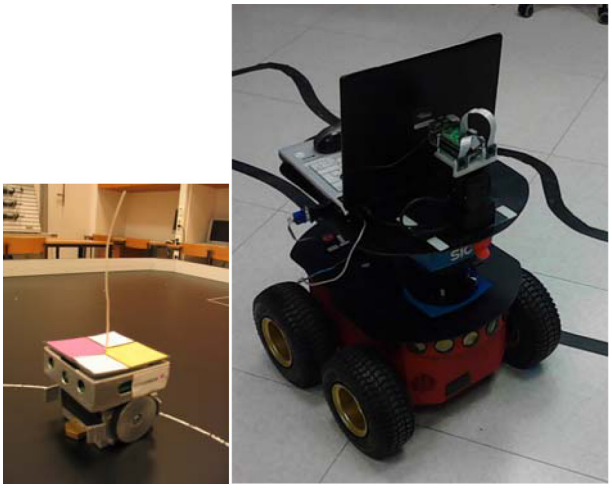

Fig. 9. Small mobile robot (left) and Pioneer 3AT robot (right) used in experiments.

4. Experimental results

In the experiments the proposed continuous model-based predictive control is compared to the discrete predictive control presented in [22]. Two mobile robot platforms are used in experiments, a smaller two-wheeled mobile robot and four-wheeled Pioneer 3AT mobile robot (see Fig. 9). Both robot motion can be approximated by differential kinematics but with different parameters. As already explained in Section 2 (comment of Eq. (1)) the controller can also be applied to Ackermann robot type using simple velocities transformations. This covers majority of wheeled mobile robots used in practice.

The small robot is designed for robot soccer competitions where speed, robustness and accuracy are needed. It fits in a cube with a 7.5-cm side and weighs 0.5 kg. The robot pose is estimated with an image sensor and a computer-vision algorithm running at $T_s = 0.033$ s sampling. The maximum allowed tangential velocity and angular velocity were $v_{MAX} = 1$ m/s and $\omega_{MAX} = 15$ rad/s, while the maximum allowed tangential wheel acceleration was $a_{MAX} = 3$ m/s^2. Pioneer 3AT is all-purpose outdoor mobile robot used mainly for research. It uses laser range finder running with 10 Hz ($T_s = 0.1$ s) for its localization. Maximum allowed velocities are set to $v_{MAX} = 0.8$ m/s and $\omega_{MAX} = 5$ rad/s.

The optimal continuous feedback control law is derived in (29) and by taking the first two rows of $(H^T T_Q H + T_R)^{-1} H^T T_Q (F_r - F)$ the gain matrix $K_c(t)$ is defined for the applied control as follows

$$u_b(t) = K_c(t)\bar{e}(t), \tag{32}$$

where $\bar{e}(t)$ is the undelayed system-tracking error, which is not available in practice due to the different sources of system delay.

Table 1
Performance of continuous and discrete control for the simulated scenarios evaluated by the root-sum-square of the pose errors and the standard deviations of the controls.

Experiment	Method	RSS_x [m]	RSS_y [m]	RSS_θ [rad]	NSS [m]	σ_v [m/s]	σ_ω [rad/s]
Correct sampling	CMPC	0.033	0.024	0.55	0.04	0.002	0.008
$T_s = 0.033$	DMPC	0.073	0.017	1.24	0.07	0.002	0.008
Double sampling	CMPC	0.021	0.025	0.24	0.035	0.003	0.016
$T_s = 0.066$	DMPC	0.068	0.030	1.07	0.075	0.005	0.018
Variable sampling	CMPC	0.110	0.210	93.6	0.23	0.002	0.009
T_{sTrue}	DMPC	0.123	0.214	94.9	0.25	0.059	0.064
variable T_{sTrue}	CMPC	0.112	0.233	85.9	0.26	0.024	0.052
and var. delay	DMPC	0.130	0.269	93.2	0.30	0.086	0.099

The main delay source of the smaller robot is an image-based sensor delay where the current camera image needs to be precessed to obtain the robot pose. The other sources causing an additional system delay are: the control algorithm computational time and the communication delay. The overall delay is nondeterministic, where $T_D = 2T_s$ is its estimated mean value. Delay of the Pioneer robot is less than T_s and is not compensated in the control.

The undelayed tracking error $\bar{e}(t)$ can be estimated from the delayed system output $e_d(t) = e(t - T_d)$ and the simulated system outputs $e_m(t)$ and $e_m(t - T_d)$ using the Smith predictor scheme as follows

$$\bar{e}(t) = e_d(t) + e_m(t) - e_m(t - T_d), \tag{33}$$

which in the frequency domain reads

$$\bar{e}(s) = e_d(s) + (sI_3 - A)^{-1}Bu(s) - (sI_3 - A)^{-1}Be^{-sTd}u(s), \tag{34}$$

inserting (34) into the control law(32) defines the control input for the delayed system as follows

$$\begin{aligned} u_b(s) &= K_c\bar{e}(s) \\ &= K_c e_d(s) + K_c(sI_3 - A)^{-1}B(I_2 - e^{-sTd})u_b(s) \end{aligned} \tag{35}$$

where I_j is the identity matrix of dimension j. The optimal controller transfer function for the delayed system then reads

$$C(s) = \frac{u_b(s)}{y_d(s)} = \left(I_2 - K_c(sI_3 - A)^{-1}B(I_2 - e^{-sTd})\right)^{-1}K_c. \tag{36}$$

The same reference-trajectory and control-design parameters as selected in simulation section are used for the smaller robot. While the reference trajectory of the Pioneer robot is $x_r(t) = 1.4\sin\left(\frac{2\pi t}{50}\right)$, $y_r(t) = 1.4\sin\left(\frac{4\pi t}{50}\right)$ and design parameters for CMPC are: $T_h = 4T_s$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, \qquad R = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix},$$

$$A_r = \begin{bmatrix} -3 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & -3 \end{bmatrix}.$$

The control parameters for both control laws (CMPC and DMPC) are selected equivalently to have the same performance.

The trajectory-tracking results (for the small robot), obtained using the proposed continuous model-predictive controller (CMPC) and comparison to discrete model-predictive controller (DMPC), are shown in Fig. 10. The trajectory-tracking results of both approaches are of approximately similar quality. Both result in good tracking in the presence of the system delay and noisy sensor data, with a standard deviation of approximately 2 mm for position and 1° for orientation. During the experiments sensor disturbances, such as the wrong pose estimation (outliers; 2% of all measurements) and some camera distortion (perspective and radial distortion) are also present.

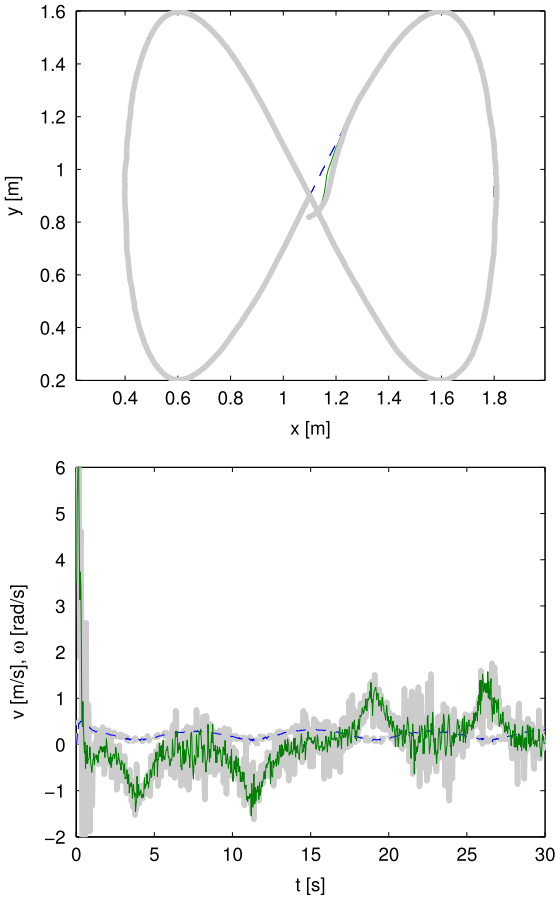

Fig. 10. Trajectory tracking experiment of the small robot with the CMPC (thin) and DMPC (thick) at variable sampling and control delay. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

A closer comparison of trajectories in Fig. 10 reveals slightly better tracking results for the CMPC during the initial transition. This is also seen in the comparison of the quality indexes in the first row of table Table 2. The main difference between both approaches is observed by comparing velocity inputs. A much higher jitter in the tangential and angular velocity is present in the discrete model predictive control, which is also seen from the standard deviations of the control variables in the first row of table Table 2. The latter statement was observed in experiment as a much smoother motion of the robot platform in the case of the CMPC.

More detailed validation is done for the Pioneer robot using constant sampling, double sampling and variable sampling as follows in Figs. 11–13 and in Table 2. From figures and table of performances the same conclusions can be drawn as in Section 3.

Table 2
Performance of CMPC and DMPC for experiments on real robots evaluated by the root-sum-square of the pose errors and the standard deviations of the controls.

Experiment	Method	RSS_x [m]	RSS_y [m]	RSS_θ [rad]	NSS [m]	σ_v [m/s]	σ_ω [rad/s]
Small robot							
sampling	CMPC	0.31	0.83	3.27	0.88	0.032	0.53
$T_s = 0.033$	DMPC	0.32	0.90	3.88	0.95	0.055	0.78
Pioneer robot							
	CMPC	0.66	1.24	1.14	1.41	0.039	0.038
$T_s = 0.1$	DMPC	0.99	1.50	1.11	0.95	0.051	0.105
Pioneer robot							
	CMPC	0.53	0.97	1.04	1.11	0.027	0.036
$T_s = 0.2$	DMPC	8.35	5.88	4.46	10.21	0.315	1.383
Pioneer robot							
	CMPC	0.61	1.1	1.01	1.26	0.041	0.042
variable T_s	DMPC	3.10	4.67	3.48	5.61	0.258	0.812

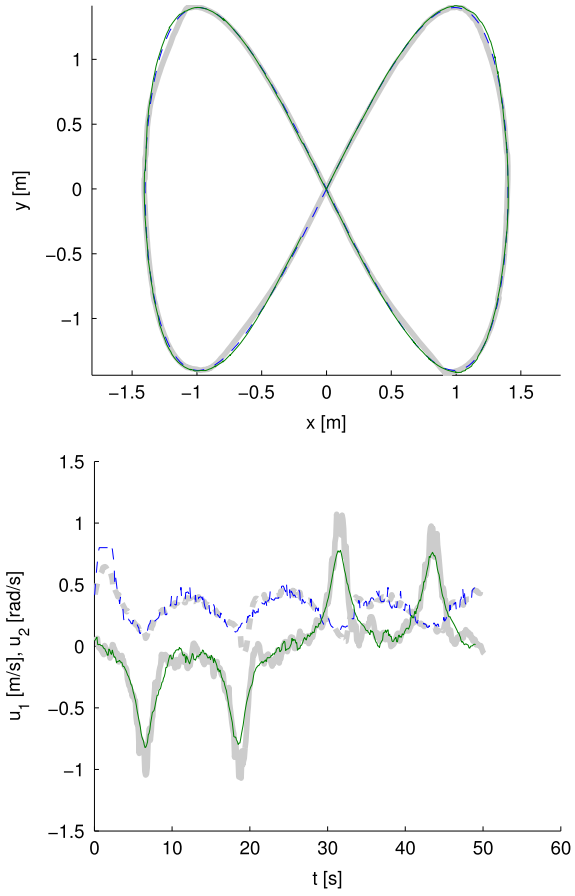

Fig. 11. Trajectory tracking experiment of Pioneer robot with the CMPC (thin) and DMPC (thick) at regular sampling time $T_s = 0.1$ s. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

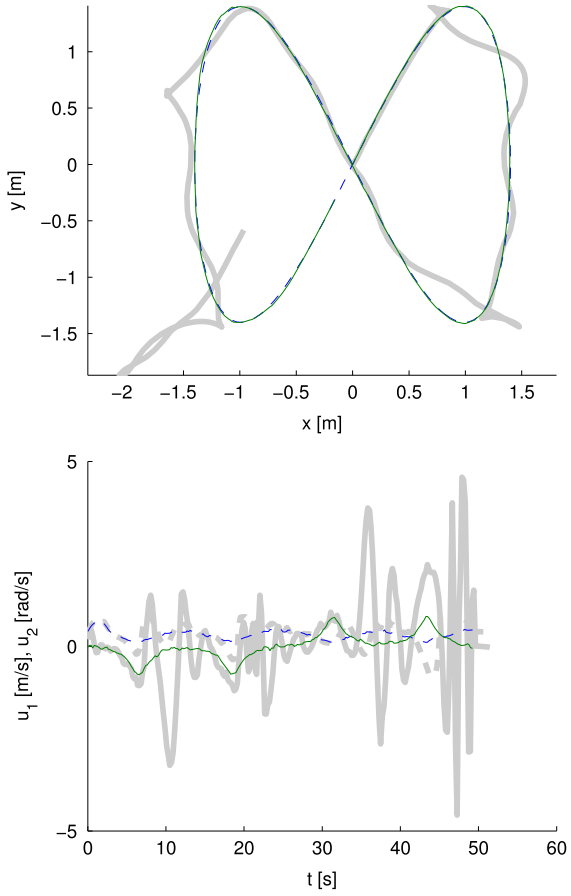

Fig. 12. Trajectory tracking experiment of Pioneer robot with the CMPC (thin) and DMPC (thick) at double sampling time ($T_s = 0.2$ s) and control design parameters tuned to $T_s = 0.1$ s. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

Continuous and discrete approaches are equivalent at regular sampling time where actual sampling time is the same as the one selected in the tuning phase (see Fig. 11). If sampling time is changed (in Fig. 12 is doubled so $Ts = 0.2$ s) and the control design parameters are not adapted (they are valid for $Ts = 0.1$ s) then the performance of DMPC becomes worse while the CMPC performance is not affected noticeably. Similarly if sampling time is changing randomly then the performance of DMPC becomes worse while the CMPC is insensitive to the sampling time period variation (in Fig. 13 sample is lost with 50% probability).

From the above comparisons the CMPC approach gives better results, which is to be expected because of the varying sampling times, mostly due to the variable control delay. The CMPC algorithm is derived in continuous space and therefore the discretization of the continuous system tracking model (7) is not needed, as it is in the case for the DMPC. This statement is also consistent with the simulation analysis made.

In the DMPC the tracking-error model discretization according to the desired sample time is made. Because the actual sampling time instants are nondeterministic, the error due to the discretization is then propagated over the whole control-law algorithm,

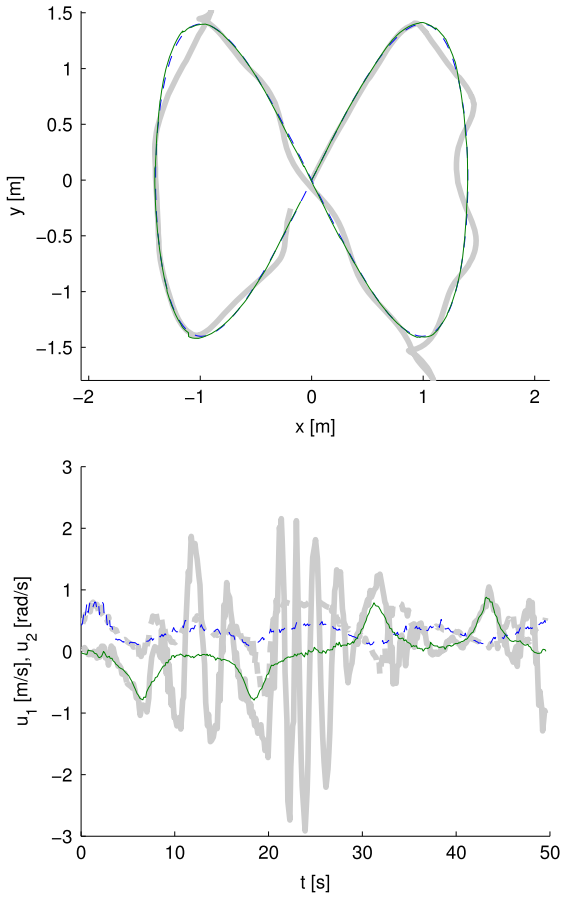

Fig. 13. Trajectory tracking experiment of Pioneer robot with the CMPC (thin) and DMPC (thick) at variable sampling time where sample is lost with 50% probability. First figure: robot path (—), reference path (- -), second figure: tangential velocity v (- -) and angular velocity ω (—).

while in the CMPC the continuous control signal is only evaluated in actual discrete time samples at the end of each control-loop iteration before sending the velocity commands to the robot platform.

5. Conclusion

The continuous model-predictive trajectory-tracking control of a mobile robot is presented in this paper. The proposed control law minimizes the quadratic cost function consisting of tracking errors and control effort as is also the case in the discrete version. The solution to the control is derived analytically, which enables fast, real-time implementations. The proposed continuous model-predictive control was validated by simulation and also on a real mobile robot.

Continuous model-predictive control design has, in ideal situations, similar performance to the equivalent discrete model-predictive control. However, in general situations the assumption of having a uniform sampling time and a deterministic control delay is not always realistic. It has been shown that a continuous design gives better results in cases where the sampling-time instants are not deterministically periodic. An important advantage of the proposed continuous model predictive control is also the better robustness of its control-law design parameters according to the sampling period. The change in the sampling period does not affect the control quality.

References

[1] I. Kolmanovsky, N.H. McClamroch, Developments in Nonholonomic Control Problems, IEEE Control Syst. 15 (6) (1995) 20–36.
[2] C. Canudas de Wit, O.J. Sordalen, Exponential stabilization of mobile robots with nonholonomic constraints, IEEE Trans. Automat. Control 37 (11) (1992) 1791–1797.
[3] K. Park, H. Chung, J.G. Lee, Point stabilization of mobile robots via state-space exact feedback linearization, Robot. Comput.-Integr. Manuf. 16 (2000) 353–363.
[4] A. Balluchi, A. Bicchi, A. Balestrino, G. Casalino, Path tracking control for Dubin's cars, in: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, 1996, pp. 3123–3128.
[5] N. Sarkar, X. Yun, V. Kumar, Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot, Int. J. Robot. Res. 13 (1) (1994) 55–69.
[6] A. Luca, G. Oriolo, Modelling and control of nonholonomic mechanical systems, in: J. Angeles, A. Kecskemethy (Eds.), Kinematics and Dynamics of Multi-Body Systems, Springer-Verlag, Wien, 1995.
[7] G. Oriolo, A. Luca, M. Vandittelli, WMR control via dynamic feedback linearization: Design, implementation, and experimental validation, IEEE Trans. Control Syst. Technol. 10 (6) (2002) 835–852.
[8] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi, A stable tracking control method for an autonomous mobile robot, in: Proceedings of the 1990 IEEE International Conference on Robotics and Automation, vol. 1, Cincinnati, OH, 1990, pp. 384–389.
[9] C. Samson, Time-varying feedback stabilization of car like wheeled mobile robot, Int. J. Robot. Res. 12 (1) (1993) 55–64.
[10] F. Pourboghrat, M.P. Karlsson, Adaptive control of dynamic mobile robots with nonholonomic constraints, Comput. Electr. Eng. 28 (2002) 241–253.
[11] F.M. Raimondi, M. Melluso, A new fuzzy dynamics controller for autonomous vehicles with nonholonomic constraints, Robot. Auton. Syst. 52 (2005) 115–131.
[12] D.-H. Kim, J.-H. Oh, Tracking control of a two-wheeled mobile robot using input–output linearization, Control Eng. Pract. 7 (3) (1999) 369–373.
[13] T.C. Lee, K.T. Song, C.H. Lee, C.C. Teng, Tracking control of unicycle-modeled mobile robots using a saturation feedback controller, EEE Trans. Control Syst. Technol. 9 (2) (2001) 305–318.
[14] S.P.M. Noijen, P.F. Lambrechts, H. Nijmeijer, An observer-controller combination for a unicycle mobile robot, Int. J. Control 78 (2) (2005) 81–87.
[15] S. Blažič, A novel trajectory-tracking control law for wheeled mobile robots, Robot. Auton. Syst. 59 (11) (2011) 1001–1007.
[16] Kühne, W.F. Lages, J.M. Gomes da Silva Jr., Gomes model predictive control of a mobile robot using linearization, in: Proceedings of Mechatronics and Robotics 2004, Aachen, Germany, 2004.
[17] A. Ollero, O. Amidi, Predictive path tracking of mobile robots. Application to the CMU Navlab, in: Proceedings of 5th International Conference on Advanced Robotics, Robots in Unstructured Environments, ICAR'91, vol. 2, Pisa, Italy, 1991, pp. 1081–1086.
[18] J.E. Normey-Rico, J. Gomez-Ortega, E.F. Camacho, A Smith-predictor-based generalised predictive controller for mobile robot path-tracking, Control Eng. Pract. 7 (6) (1999) 729–740.
[19] H.A. van Essen, H. Nijmeijer, Nonlinear model predictive control for constrained mobile robots, in: European control conference: ECC '01, Porto, Portugal, 2001, pp. 1157–1162.
[20] H. Hu, D. Gu., Neural predictive control for a car-like mobile robot, Robot. Auton. Syst. 39 (2) (2002) 73–86.
[21] S. Adinandra, E. Schreurs, H. Nijmeijer, A practical model predictive control for a group of unicycle mobile robots, in: 4th IFACNonlinear Model Predictive Control Conference, vol. 4, No. 1, Leeuwenhorst, Netherlands, 2012, pp. 472–477.
[22] G. Klančar, I. Škrjanc, Tracking-error model-based predictive control for mobile robots in real time, Robot. Auton. Syst. 55 (6) (2007) 460–469.
[23] E. Kayacan, H. Ramon, W. Saeys, Robust trajectory tracking error model-based predictive control for unmanned ground vehicles, EEE/ASME Trans. Mechatronics 21 (2) (2016) 806–814.

[24] M.F. Sagfors, H.T. Toivonen, H infinity LQG control of asynchronous sampled-data systems, Automatica 33 (9) (1997) 1663–1668.
[25] M. Wargui, M. Tadjine, A. Rachid, A. Stability of real time control of an autonomous mobile robot, in: 5th IEEE International Workshop on Robot and Human Communication, 1996, pp. 311–316.
[26] H. Gao, T. Chen, Stabilization of nonlinear systems under variable sampling: A fuzzy control approach, IEEE Trans. Fuzzy Syst. 15 (5) (2007)
972–983.
[27] G. Dudek, M. Jenkin, Computational Principles of Mobile Robotics, second ed., Cambridge University Press, 2010.
[28] A. Luca, G. Oriolo, C. Samson, Feedback control of a nonholonomic car-like robot, in: J.P. Laumond (Ed.), in: Lectures Notes in Control and Information Sciences, vol. 229, Springer-Verlag, London, 1998, pp. 171–253.
[29] A. Luca, G. Oriolo, M. Vendittelli, Control of wheeled mobile robots: An experimental overview, in: S. Nicosia, B. Siciliano, A. Bicchi, P. Valigi (Eds.), RAMSETE - Articulated and Mobile Robotics for Services and Technologies, Springer-Verlag, 2001.
[30] C. Samson, K. Ait-abderrahim, Feedback control of a nonholonomic wheeled cart in Cartesian space, in: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, vol. 2, Sacramento, CA, 1991, pp. 1136–1141.
[31] R.W. Brockett, Asymptotic stability and feedback stabilization, in: R.W. Brockett, R.S. Millman, H.J. Sussmann (Eds.), Differential Geometric Control Theory, Birkhuser, Boston, MA, 1983, pp. 181–191.
[32] J. Nilsson, Real-Time Control Systems with Delays, Department of Automatic Control Lund Institute of Technology, Lund, 1998.
[33] K.J. Aström, B. Wittenmark, Computer-Controlled Systems, Theory and Design, Prentice Hall, 1996.
[34] C. Purcaru, R.-E. Precup, D.T. Iercan, R.C. David, Hybrid PSO-GSA A robot path planning algorithm in static environments with danger zones, in: 17th International Conference: System Theory, Control and Computing, ICSTCC, 2013, pp. 434–439.

Igor Škrjanc received the B.Sc., the M.Sc. and the Ph.D. degrees in electrical engineering, in 1988, 1991 and 1996, respectively, from the Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia. His main research interests are in adaptive, predictive, fuzzy and fuzzy adaptive control systems. In 2007 he received the highest research award of the Faculty of Electrical Engineering, Vodovnikova award and in 2008, award of the Republic of Slovenia for Scientific and Research Achievements, Zois award for outstanding research results in the field of intelligent control. He also received the Humboldt Research Fellowship for Experienced Researchers for the period between 2009 and 2011 for the research work at University of Siegen.

Gregor Klančar received B.Sc. and Ph.D. in electrical engineering from the University of Ljubljana, Slovenia in 1999 and 2003 respectively. He is currently employed as an assistant at the Faculty of Electrical Engineering at the University of Ljubljana. His research work focuses on the area of fault diagnosis methods, on the mobile robotics area and on the area of control and supervision of multiagent systems.